



Citation for published version:

Joyce, S & Harding, J 2013, The Generator 2.0. in *Rethinking Prototyping: Proceedings of the Design Modelling Symposium Berlin 2013*. Epubli.

Publication date:

2013

Document Version

Early version, also known as pre-print

[Link to publication](#)

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

The Generator 2.0

Sam Conrad Joyce,
University of Bath, United Kingdom

John Harding,
University of Bath, United Kingdom

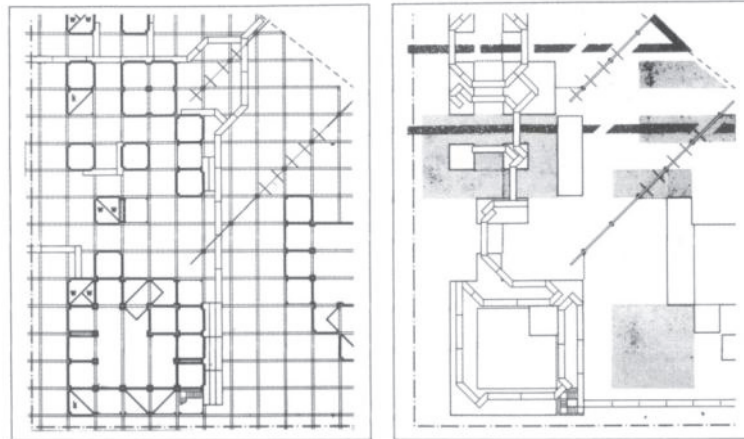


Figure 1: Plan showing an arrangement of the original Generator (Price 1984)

INTRODUCTION

It has been over 30 years since the original conception of the ‘Generator’ by Cedric Price. A project that in his own words was “little to do with problem solving, rather creating the desirable conditions and opportunities hitherto thought impossible” (Price 1984). The building aimed to develop a dialogue with its users, by acting as a living working prototype continually reconfiguring in a dynamic conversation, in complete contrast to architecture as monument. The Generator was to be a truly complex adaptive machine, with local system rules and initial condition established but with no fixed global goals and requirements over time.

This paper postulates that the ideas behind the generator as a ‘thinking’ adaptive building are important and worth investigating and progressing. To begin, we introduce the original design and then go on to highlight the conceptual divide between the desired system and its actual real capability. From this study, two particular areas of development for a ‘Generator 2.0’ project are identified: distributed learning and temporal responsiveness.

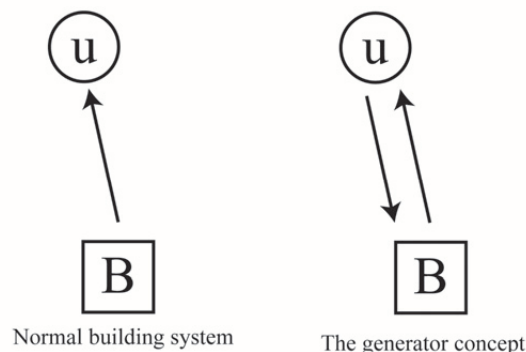


Figure 2: A systems diagram of building ‘B’ and user ‘u’.

Feedback and Learning

The Generator was intended to learn from its users. Learning in this context is defined as the modification of both the building and its users via feedback (Fig. 2). This cycle exists with normal design processes, whereby designer’s actions

are modified by experience of the performance of previous designs as well as reflection on their own process. This is a direct example of ‘Second-Order Cybernetics’ and has been shown to be required for a flexible system (Von Foerster 1981). The proposed learning ability of the original Generator was confined to basic artificial intelligence methods (Michie 1974), modifying variables of the system but not the topological structure of relationships.

Feedback, learning and reconfiguration all importantly exist in the temporal dimension. It is the authors view that this was an under explored facet in the original system, even though it made breakthrough steps by involving a concept of boredom where the system acted proactively if underused. Demonstrating a different approach both at a conceptual and pragmatic level which is able to act and learn over time is the core aim of the paper as it is believed that this breaks many original constraints and enables a much wider range of integration between user and building, thus establishing a conversation with each other in a autopoietic structural coupling (Maturana and Varela, 1979)

With this approach in mind, the latter part of the paper gives a concrete example application of a proposed Generator 2.0 concept, acting as a true distributed learning system. This project is based on the input of distributed sensors from each of the units, using local information to modify its behaviour. In this way it is possible to conceive of a building system able to modify itself at a systematic level, hence such the Generator 2.0 has the potential to realise the original goal of building as its own autonomous architect, developing its own concepts for objects as part of a continual developmental process.

THE ORIGINAL SYSTEM

Background

Developed in the late 1970s and planned to be realised at the White Oak Plantation in Florida, The Generator project was conceived as a reconfigurable multiple activity building for small groups of both public and private users. The physical manifestation was a basic modular system made of movable walls and sliding panels positioned on a regular grid of foundations. This kit of parts could then be reconfigured according to temporal user requirements, with a crane system implemented similar to perhaps Price’s most famous work, the Fun Palace project conceived ten years earlier.

A Cybernetic system

The significant innovation of the Generator was to explore the intent and possibilities of what a highly reconfigurable structure paired with a sophisticated computational system could enable. Now working as systems designer, Price turned to the cybernetician and long term collaborator Gordon Pask in order to establish a suitable system design for the project. As Pask remarked himself, “architecture is a discipline that fits the bill insofar as the abstract concepts of cybernetics can be interpreted in architectural terms” (Frazer, 2001). The resulting system (Fig. 3) was a fixed systems model that included designated human participants (named ‘Polariser’ and ‘Factor’ by Price) charged with rearranging the layout according to the system rules and user requirements which were entered at each iteration.

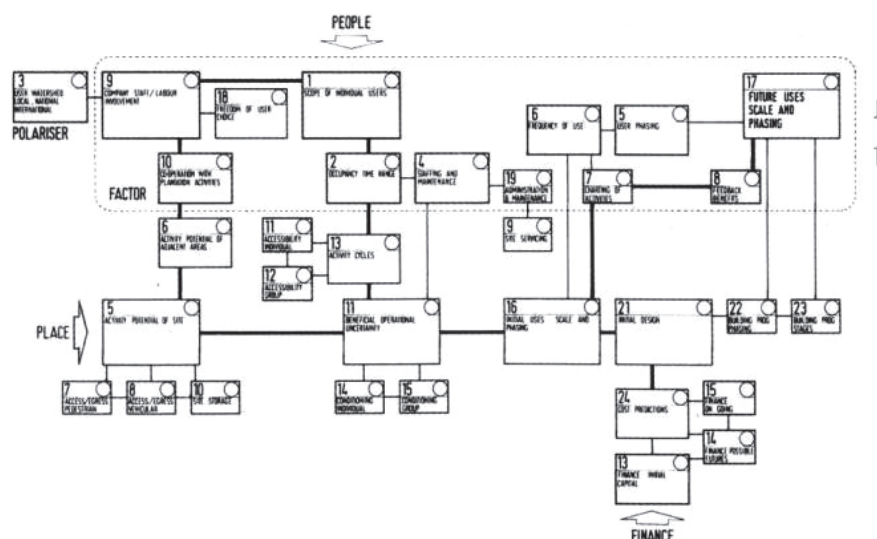


Figure 3: Gordon Pask's proposed cybernetic system for the Generator project (Price 1984).

Integrating the machine

Later in the project it became clear to Price that the system did not require humans to follow systematic rules to rearrange the building and that the use of computers (in their infancy at the time) could replace 'Polariser' and 'Factor' as well as provide interesting new possibilities for the project. He therefore turned to the architects/programmers John and Julia Frazer to assist.

In keeping with Price's anti-architect approach, the computer was intended to replace the human completely in terms of configuring the Generator. John and Julia therefore set about creating a "set of programs that would mimic a team of architects" (Stansell 1981). Four separate computer programs were developed that would use input from such sensors. The first three provided what Price called the 'perpetual architect' drawing program that held the data and rules for Generator's design. Using this in-built system, the computer could make new suggestions based on user needs, offering potential layout configurations before committing the design for construction, and hence allowed further interaction with its users (Frazer 1979 cited in Steenson 2010). As well as automating the Generator system, technology was discussed such as sensor devices located in the Generator components (such as walls and screens) offering automated feedback to the building system as opposed to the manual 'turn-based' user entry originally proposed by Price. By embedding the electronics in the components themselves, it was envisaged that the whole building could have acted as a processor distributed throughout the whole building with electronics embedded in each component (Frazer 1997).

Computer gets bored

At this point in the project, although the generator was able to offer novel configurations for its users, the rules of the system were still 'designed' by Price. However, further design iterations explored the possibility of Generator being able to generate its *own* layout proposals if it became 'bored'. The Generator would "continually try to stimulate proposals for change. If the buildings do not receive any response, the computer declares itself bored and produces unsolicited plans for the occupants to consider anyway" (Stansell 1981). John and Julia Frazer therefore assisted in developing the software to enable random re-configuration should the layout remain in stasis for a certain amount of time.

Here the building/computer has the potential to 'surprise' its users and create unexpected interactions outside of what traditional architectural practice would create (Steenson 2010). Conceptually, this was an interesting step for the project, Price had extended his 'anti-architect' design ethos even further than before, conceiving of a building not made from a single condition imposed from the onset by the designer (Fig. 4) but instead towards an environment able to respond to and create "calculated uncertainty" (Price 1984).

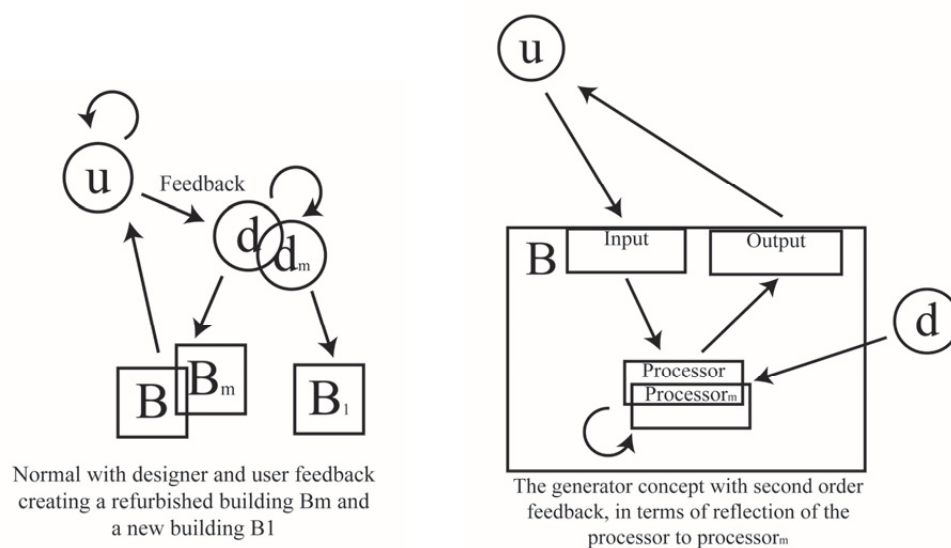


Figure 4: Diagram showing the process of learning in both the typical design process (left) and the generator system (right), learning is shown as a modification of the designer or processor respectively by the suffix 'm'. The designer 'd' is shown in both diagrams to highlight the difference in role.

The learning process

The original idea proposed by Price was intended to learn over time according to an in-built measure of success. This learning depended on an algorithm that employed stochastic distributions to learn favourable output configurations inspired by the work of Donald Michie (1974). Good configurations would therefore be reinforced over time, however the measure of 'goodness' remained embedded in the system itself and was not capable of change. In any effect, the actual implementation tested by John and Julia Frazer did not include such a system and remained an undeveloped concept.

THE ACTUAL IMPLEMENTATION

Proof of concept

Although the Generator project suggested an interesting approach to design and backed by a wealthy client, unfortunately like many of Price's projects it remained unrealised. However during the project, a physical proof of concept implementation was conducted carried out by John and Julia Frazer. The physical system developed linked the physical arrangement of model units with a computing system working as a parallel machine (Fig. 5). Such a device allowed the position of the units' to be understood by the computer and visualisations created from global performance metrics such as gross floor area, material costs, and solar performance. This live system allowed users to hone and propose their designs, via intuitive physical modelling units, augmented by higher levels of analysis not then typically accessible to the non-professional.

The development of the generator was exceptionally sophisticated for the time, and is still in development by John Frazer by "considering the whole city as a Generator" (John Frazer, pers. comm., 2013). Similarly it is the opinion of the authors that with recent advances in modern technology, the original project is ready to be re-visited and extended upon.

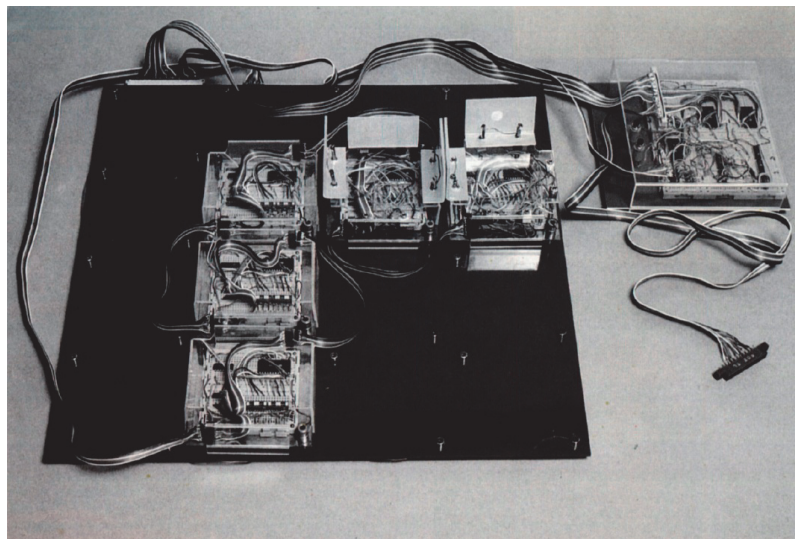


Figure 5: Physical model of the developed Generator system (Frazer 1995).

An under-developed idea

The original project proposed interesting design ideas, but the reality was that the project's development never finalised in scope probably because each new avenue offered exciting new possibilities for Price. Ironically, like the intended building the design itself became an ever changing project.

The learning system inspired by Michie whilst proposed was never tested or formalised in any way. Had it been attempted, the rigid 'top-down' cybernetic system as originally proposed by Price and Pask would have made any adaptation difficult for the generator in an uncertain environment as the system itself was required to be designed *a priori* and embedded in the computational systems as developed by Frazer and Frazer.

The intended learning system had an internal measure of success and therefore incorporated no user feedback, only desired requirements at the front end. Although the computer program could sense when to suggest an alternative by

having a memory of previous states, these states were not used as part of an artificial learning system *over time*, that is, they could not adapt to changes in what constituted a *successful* implementation.

As Frazer comments, “ultimately, the building itself might be better able to determine its arrangement for the users’ benefit than the users themselves”, citing the example of environmental control systems with a learning capacity (Frazer 1995). However, one must be careful to distinguish how the ‘benefit’ is measured. If the metric remains embedded in the system then the user becomes separated from the machine, much like environmental control systems are often ‘optimised’ to a state imposed externally and not by the users themselves. An everyday example may be that of an air-conditioned train that adjusts the carriage environment to a pre-programmed global state and not to the particular on-board users’ requirements at the current moment in time.

To summarise, having a top-down system/model that was designed externally with fixed goals and unable to be modified would be unable to sufficiently adapt to the complexity of human behaviour, which is in a constant state of flux. Here an opportunity for implementing additional dynamic user feedback in an adaptive system could be possible for Generator 2.0., acknowledging principles in second-order cybernetics whereby any observer of the system is also inherently part of it.

Time scales

One under-developed dimension in the original system as intended and implemented is that of time. The original system was essentially that of a turn based interaction, between the building program and the user. As noted before, the original proposal included the concept of boredom, which produced unsolicited designs for the user to consider if the system was not interacted with over a given time. Whilst revolutionary this was the only demonstrated proactive capability of the actual system implemented in the model.

Instead, what might be required is the ability for the system to establish a concept of time by its own. One example of this is the work of Turing’s reaction diffusion model (Turing 1952), where various actions (diffusion) are initiated by various interacting states (reaction). This method would be able to dynamically respond to users, and if combined with the principles of reconfigurable higher order processes it has the potential to produce a truly responsive adaptive system. As human behaviour is cyclic in nature (daily, weekly, monthly, etc...), enabling the Generator to be able to synchronise and hence learn certain configurations for certain times is an interesting prospect.

With these thoughts in mind, it therefore seems appropriate to look to self-organising systems with no fixed topological logical structure for any new implementation of the Generator. Such an approach is discussed in the next section.

GENERATOR 2.0

An implementation of the Generator with a dependence on learning and interaction with users requires a suitable freedom and complexity itself. This was the main contradiction of the original design that although claiming to be an anti-architect, Price was still imposing operational behaviour *explicitly* at the systematic level (Fig. 4), and this system structure (or network) was essentially fixed and incapable of adaptation and learning over time.

An implicit approach

One approach to freeing up the system is to enable the cybernetic control network with topological flexibility. Such an approach is analogous to genetic programming (Koza 1994) with networks instead of tree structures. This has been investigated in an architectural context by Coates et al. (1999) by combining Lindenmayer Systems with genetic algorithms and more recently by Harding et al. (2012) in relation to evolving associative models for architectural geometry with no fixed body-plan. It is worth considering for the latter, that whilst human cognition of an evolved explicit network structure offered benefits, in terms of Generator 2.0, a conversation between building and users can take place without a complete understanding of the black box inside the machine. The users do not need to understand *how* the building/machine is thinking in order to form a relationship with it.

Instead, inspired by the work of Frazer (1995) and Coates et al. (1996), the use of a cellular automaton (CA) combined with a genetic algorithm provides sufficient complexity combined with a learning system. The self-organising nature of the CA thus allows for morphogenesis at the level of the system with no fixed body-plan. In their paper ‘Three ways to grow designs’, Bentley and Kumar state that an implicit embryogeny such as a cellular automaton (CA) is best able to explore designs due to its un-reliance on an internal structure, hence preferred to evolving an explicit network (Bentley; Kumar, 1999) When combined with a genetic algorithm, the CA’s low level rules were evolved to solve a simple design task. Importantly, Bentley and Kumar found that the CA was also able to adapt to various scales, and hence work as an

‘organised’ complex system operating at different hierarchical levels (Simon 1962). In terms of Generator 2.0, this is crucial in order to develop learning across various time and spatial patterns as discussed in the previous section.

‘Understanding’ the machine

Unlike a top-down explicit network, a self-organising system such as a cellular automaton is very difficult to reduce, and hence understand due to its inherent complexity. *How* the machine works no longer becomes cognisable by humans. Unlike the original systems network set out by Price and Pask, Generator 2.0 has no such fixed system, instead low level behavioural rules much closer to the original concept of building as parallel processing machine.

Having no explicit structure at the level of the system therefore gives rise to flexibility at a higher level of abstraction, however one must give up an understanding of process unless running the whole system again from its initial state. In the context of the Generator project, it is certainly not practical for the users of the building to attempt to reduce the building/machine to this level of scrutiny.

Fortunately, much like a conversation between human and human, there is no need for the other to understand how one is *thinking* (or indeed describe human consciousness) in order to establish a structural coupling and form a symbiosis. There is no need, for example, for a flower to understand the inner workings of a bee’s mind in order to form a beneficial (for both parties) relationship with it. Thus we should acknowledge that it might be impossible to untangle the mess and understand how the Generator 2.0 works at every stage of its development.

This means that rather than “mimic” a team of architects, the Generator 2.0 begins to establish its own concepts and behaviours that may not be similar to anything human. In order to engage in a conversation, the building elements and the spaces in-between act as a language of communication between Generator 2.0 and its users.

Implementation

Due to advances in processing power, computer simulation can offer clues as to whether a Generator 2.0 project could be successful. Although it would be impossible to predict the environment for which the project would exist in, making sure the learning aspects of the projects could function under varied circumstances would indicate certain robustness in a changing environment.

The basic system developed was intended as a simplified but ultimately practical and realisable version of the Generator with all the important discussed differences included. It comprises of two interacting groups; the cells and the agents. The cells represent the building and mirror the Generator but reduced to the function of a basic roof or shading device with only an ‘on’ or ‘off’ condition representing a covered or uncovered state. These cells are arranged in a grid and so can be regarded in much the same way as a cellular automata (CA). The agents are added to represent the users in the building, whilst modelled here for the benefit of the simulation, it is important to note that this role is intended for real human users and whilst the behaviours of the agents is relatively simple, the emphasis is on the cells’ ability to scale their own complexity up to respond to the users behaviours.

The scenario is a repeated ‘day’ of agent inhabitation of the space, where agents carry out the same routine but initiated under a different random seed, this is an important feature as it shows the system’s ability to develop and adapt its thinking to the users over a time frame.

Process	State				Transform	
	External	Internal				
	#1	#1	#2		#1	#2
Rule 1	= 12	> 1	> 5		= 0	+ 10
Rule 2	< 20	= 1	ignore		× 2	÷ 5
Initial State: Cell State (t)	12	1	45			
Test Rule 1	12=12 pass	1 > 1 fail	45 > 5 pass			fail
Test Rule 2	12<20 pass	1 = 1 pass	~ pass			triggered
		1*2 → 2	45/5 → 9			apply transform
Next State: Cell State (t+1)	12	2	9			
Test Rule 1	12=12 pass	2 > 1 pass	9 > 5 pass			triggered
		2 → 0	9+10 → 19			apply transform
Test Rule 2	12<20 pass	0 = 1 fail	~ pass			triggered
Next State: Cell State (t+2)	12	0	19			

Figure 6: Example of two cell iterations involving processing two rules and three states

Genotype to phenotype

To enable the level of flexibility desired in the Generator's understanding and behaviour, a much lower level methodology for the cells was required, quite different from intentionally described actions in top down systems or predetermined emergent systems. The external (passively sensed un-controlled) and internal (active modifiable/controllable) states of the cells are represented as one state vector. Each cell then has a variable number of rules that are constantly evaluated and triggered where rules comprise of preconditions and transformations. Triggering occurs if the preconditions are satisfied by the current state vector.

These preconditions are a vector of conditionals equal in length to the internal and external states and represented as a value and a relational operator pair. A precondition is satisfied if for each state the cell-state, relational operator, condition-value equation equals true. If all are satisfied, the rule transformation is applied which is a vector of value and operator pairs that is then applied to the cells current state. An example of this process is shown for one cell in (Fig. 6). The genotype is therefore mapping from bit string to an initial condition and the rules of the CA.

Whilst at one level this represents a very basic generic system this is very similar to processes of natural cell embryogenesis but with states rather than chemicals, to activate or suppress rules rather than genes (Bentley and Kumar, 1999); however unlike Bentley and Kumar, the actual phenotype is a changing reactive CA that runs throughout the day, not just a final outcome at a certain iteration. For this project the only external state of the cell was linked to the number of agents currently in the cell, whilst the open and closed state of the shading was linked to the first internal state being of value zero and one respectively, other internal values were allowed also but had no effect on the environment.

Human feedback

System learning is achieved from a coupling of the development of these rules and their task solving, with good solutions being reinforced, whilst still allowing for chance mutations (new designs). What is of interest here is that the goal can be defined collectively by the input of the users.

In this example, the success of the cells current strategy (collection of rules) is determined by two factors; firstly the utilisation of the shaded areas (pre-determined objective), and secondly a measure of user satisfaction (subjective opinion). The latter measurement is made possible by allowing users to submit simple feedback following use (e.g. a 'like' button or similar). This may seem similar to the original Generator, however there is no explicit measure of user 'modification' as Pask's original cybernetic diagram suggested. Instead, users give feedback as to how well the building has suited their needs intuitively, but crucially not having to explain *how* it has done so.

It can be said then, that the Generator 2.0 has a desire to satisfy its inhabitants and not to disatisfy them (this fact must be set out by the authors), but *how* this is achieved is not made explicit. This area is perhaps the most delicate of the entire system, for one must not presuppose how the machine judges success, however it is perhaps safe to assume that if

the general goal of this architectural project is to satisfy its users' need in some way, then in the authors' opinion that is an assumption that can be made and something acceptable to embed in the Generator 2.0 system.

Learning Process

The system learns over time by assessing the success of its phenotype design (in terms of satisfied users) by comparing to previous phenotypes, for example the previous 30 days. A genetic algorithm with roulette wheel selection is used to ensure that relatively successful genotypes have a greater chance of survival for the next CA phenotype (Figure 8). The fitness value of previous days however has a decay, a multiplier of less than one applied to the fitness of each historical run. This enabled something approaching a memory of the previous designs that the system can refer to, but one which gradually forgets, allowing the system to use prior knowledge but still adapt to changes in user requirements as it reduces the potential of the system to fixate on a design instead promoting innovation.

The nature of a genetic algorithm as opposed to for example simulated annealing, means that it can run ad-indefinitum, similar to the continual evolutionary development of Dawkin's Biomorphs (1986). Mutations in the genotype, give rise to the possibility of large changes to the phenotype because of the nature of a complex system such as a CA. This ensures that the concept of boredom is incorporated into Generator 2.0. and complete stasis is averted. The machine will always attempt to try out something new at some point, regardless of success.

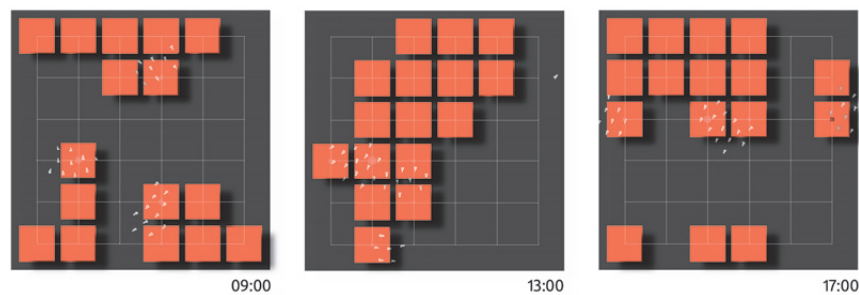


Figure 7: Example run of an effective evolved system behavior, showing shading squares covering agents roaming around in the space.

Results

A trial run on a 6x6 grid was conducted, with an agent based system simulating the behaviour of human users occupying the space for varying time periods of the day. Figure 7 shows a typical day with the CA following local rules from an initial condition.

Whilst the current setup is simple the range of behaviours generated where vast. These behaviours showed quite some variation and tended to use all the rules and states available, but most effective solutions where simple with any extra states ignored or toyed with, and most rules un-triggered, similar to junk DNA. As such an imposed upper limit of three rules per cell and two internal states was applied. Some tuning was required to set bounds, population size (number of previous days) and the level of mutation and recombination to increase the number of productive results.

After this the system was able to settle on some basic strategies to generate designs that left satisfied users but was also able to adapt if the user needs changed.. Most typically to link the cells external state of having agents in it to a internal shaded state, with some kind of mechanism to stop shading with underuse, some even developed basic counters thus evolving a it's own concept of time.

Up until this point, for this simulation we had been assuming that satisfaction was aligned with being covered by the shade for each user. However, during further simulation it was possible to invert the users'/agent's concept of satisfaction (something that had to be done top-down in this simulation trial) to be wanting not to be covered and see the system change its behaviour to accommodate. As mentioned before, in reality the reason's governing satisfaction were set by the authors in this small example, although in a real implementation, human users would give this feedback themselves without having to explain *how* they were satisfied (as an example, perhaps someone would find the shading devices beatiful to look at!)

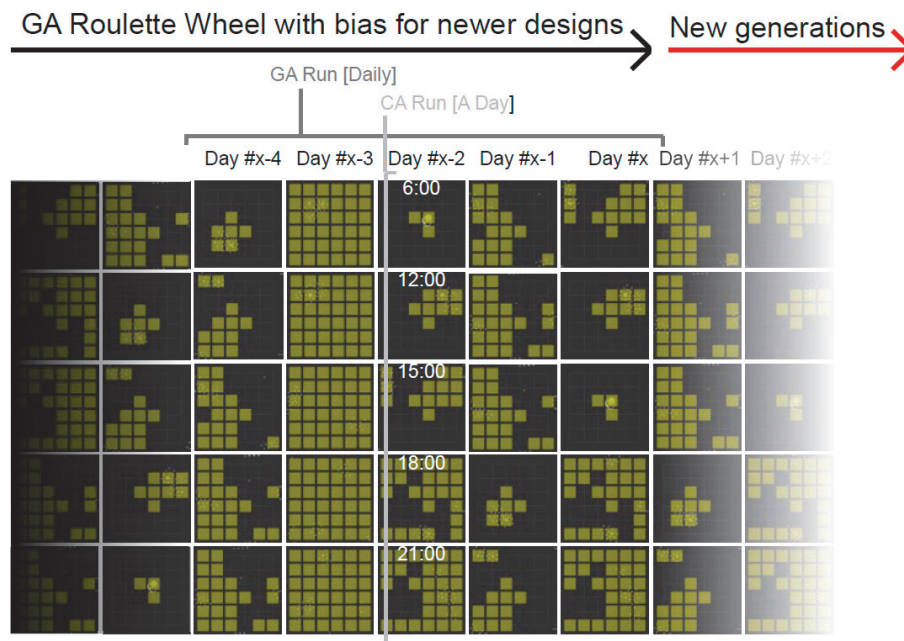


Figure 8 : Diagram showing the cycles of CA day runs against GA cycles

Discussion

Whilst these ideas are simply implemented it still created too much complexity initially, and there is a danger of unproductive over exploration of the problem. Initially something in the order of hundreds of runs were required to develop effective solutions this may not be acceptable if applying the system on a day cycle for it to take that many days to be useable. More careful selection of ranges of numbers and operators helps to vastly reduce search space and aid convergence; however this is essentially applying a wilful intentionality on the system, which is at the cost of its own creativity. It is the opinion of the authors that for future work there is much scope for the application of more sophisticated methods existing in the A.I. literature which are able to provide more economical mechanisms for learning and minimising the number of attempts required to configure meaningful logical rules.

Importantly the study shows that a building can develop its own behaviours and decision processes that amount to learning, which is radically different from typical intentionally designed systems. It proves that a system can be given initially unrelated inputs and outputs and produce working processes which relate to a user's desires implicitly rather than requiring the user to explain explicitly what it wants. Furthermore it shows that such a system can modify its own internal logic or thinking to new users desires and adapt to new situations.

CONCLUSION

The original proposition of The Generator is a strong one, however the technology and knowledge at the time was not sufficient to deliver the project as intended. Alongside the availability of modern technology, we have also argued that the development of higher level complexity (thinking/learning) is required to have truly meaningful interaction.

We have presented a theoretical approach to what is required by such a system, followed by an example of how this could be practically implemented. A new version of the Generator is therefore intended to fulfil Price's intention of a truly independent system once fully developed that can establish a structural coupling with its users. Such a machine/building system should not mimic human behaviour but develop its own character through its existence and interaction with users.

The initial application tested as a computer system gives promising results. Thus it may be possible to devise a flexible system capable of adapting and learning over time, adapting at both the physical (building) and the systematic (process) level, releasing itself from a top-down human imposed intentionality as initially proposed by Price and Pask. The next stage is to realise a physical system and allow real people to begin to interact with the Generator 2.0. Hence instead of developing fixed buildings to serve our assumed to be static requirements, we begin a converse with architectural machines as they converse with us..

REFERENCES

- Bentley, P. ; Kumar, S., 1999: Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In: *Proceedings of the genetic and evolutionary computation conference*, 1999, 1, 35-43.
- Coates, P.; Healy, N.; Lamb, C.; Voon, W., 1996: The use of Cellular Automata to explore bottom up architectonic rules *Eurographics UK Chapter 14th Annual Conference*, 26-28.
- Coates, P.; Broughton, T. & Jackson, H. 1999: Exploring three-dimensional design worlds using lindenmayer systems and genetic programming *Evolutionary design by computers*, San Mateo, CA: Morgan Kaufmann, 323-341.
- Dawkins, R. 1986: *The blind watchmaker: Why the evidence of evolution reveals a universe without design*. WW Norton & Company.
- Frazer, J. 1979: Letter to Cedric Price (January 11, 1979). Cedric Price Archives (Montreal: Canadian Centre for Architecture).
- Frazer, J. 1995: *An evolutionary architecture*. Architectural Association Publications.
- Frazer, J.H., 1997: Action and Observation: The Groningen Experiment. In: *Systemica Special Issues*, Nos 1-6, Vol 12, pp 135-142.
- Frazer J.H. 2001: The cybernetics of architecture: a tribute to the contribution of Gordon Pask. In: *Kybernetes*, 30:5, pp. 641-651.
- Harding, J.; Joyce, S.; Shepherd, P.; Williams, C., 2012: Thinking Topologically at Early Stage Parametric Design. In: *Advances in Architectural Geometry 2012*, Springer, 67-76.
- Koza, J. R., 1994: Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, Springer, 1994, 4, 87-112.
- Maturana, H.; Varela, F., 1979: *Autopoiesis and Cognition: The Realization of the Living*. Boston.
- Michie, D., 1974: *On Machine Intelligence*. Edinburgh University Press.
- Price, C., 1984: *The Square Book*, London, John Wiley.
- Simon, H. A., 1962: The architecture of complexity. *Proceedings of the American philosophical society*, JSTOR, 1962, 106, pp. 467-482.
- Stansell, J., 1981: The Building that moves in the night, *New Scientist*, Vol. 89, No. 1245, pp. 743.
- Steenson, M.W., 2010: Cedric Price's Generator. In: *Crit* 69, pp. 14-15.
- Turing, A.M. 1952: The Chemical Basis of Morphogenesis. In: *Philosophical Transactions of the Royal Society of London*, 237 (641): pp. 37-72.
- Von Foerster, H., 1981: *Observing Systems*, Intersystems Publications.